IN THE CLAIMS

A list of the pending claims is presented below. No claim amendments are included.

1. (Previously Presented) A pipelined multistreaming processor, comprising:

an instruction cache for concurrently providing a plurality of instructions for a plurality of instruction streams;

fetch logic coupled to said instruction cache enabled to concurrently fetch said plurality of instructions for said plurality of instruction streams from said instruction cache;

a plurality of instruction queues coupled to said fetch logic where each one of said plurality of instruction queues is associated with at least one of said plurality of instruction streams, wherein the number of said plurality of instruction queues is greater than said plurality of instruction streams that are provided by said instruction cache;

a dispatch stage coupled to said plurality of instruction queues for selecting and dispatching instructions for said plurality of instruction streams to a set of execution units; and

select logic coupled to said instruction cache, and to said plurality of instruction queues, said select logic monitoring each of the plurality of instruction queues, said select logic selecting ones of said plurality of instruction streams to fetch instructions from said instruction cache, the selecting based on the monitoring.

2. (Previously Presented) The processor of claim 1 wherein the number of said ones of said plurality of instruction streams selected by said select logic for fetching, is less than the number of said plurality of instruction streams.

3. (Previously Presented) The processor of claim 2 wherein said select logic monitors

a plurality of program counters (FPC), each of said plurality of program counters associated with one of said plurality of instruction streams, and wherein said select logic directs fetching from said instruction cache at addresses stored in said plurality of program counters.

4. (Previously Presented) The processor of claim 3 wherein said select logic also monitors said plurality of instruction queues and determines said ones of said plurality of instruction streams, for which to fetch, based on how full said plurality of instruction queues are.

5. (Canceled)

6. (Previously Presented) The processor of claim 1 wherein said fetch logic concurrently stores fetched instructions into ones of said plurality of instruction queues that are associated with said ones of said plurality of instruction streams fetched by said fetch logic.

7. (Previously Presented) In a multistreaming processor, the processor executing a plurality of instruction streams, a method for decoupling fetching of instructions for the plurality of instruction streams from the dispatch of those instructions for execution, the method comprising:

providing a plurality of instruction queues in the processor, one instruction queue for each instruction stream, the plurality of queues located between fetch logic, which fetches instructions for the instruction streams, and dispatch logic, which dispatches instructions for the instruction streams, each of the plurality of instruction queues queuing a plurality of instructions for its instruction stream; and

providing select logic in the processor, the select logic monitoring each of the plurality of instruction queues and selecting a plurality of instruction streams for which to fetch instructions from an instruction cache based on said monitoring, the number of instruction streams selected by the select logic being less than the number of instruction streams in the processor;

wherein by selecting a plurality of instruction streams for fetching which is less than the number of instruction streams executing, the association between fetching of instructions, and their dispatch is effectively decoupled.

8. (Previously Presented) The method of claim 7 further comprising:

having the select logic choose which of the instruction streams to fetch based on the contents of the plurality of instruction queues, the select logic choosing to fetch instructions for those instruction streams whose instruction queues are least full.

9. (Previously Presented) The method of claim 8 wherein the select logic further comprises a plurality of fetch program counters, each for storing an address associated with the next instruction to be fetched, for each of the plurality of instruction streams.

10. (Previously Presented) The method of claim 9 wherein the fetch logic utilizes the contents of the plurality of fetch program counters to fetch instructions for the selected plurality of instruction streams.

11. (Previously Presented) The method of claim 7 wherein the dispatch logic dispatches instructions to a plurality of execution units.

12. (Previously Presented) The method of claim 11 wherein the plurality of execution units comprises eight arithmetic logic units (ALUs), and two memory units.

13. (Previously Presented) The processor of claim 1 wherein the monitoring performed by said select logic determines how full each of the plurality of instruction queues are.

14. (Previously Presented) The processor of claim 13 wherein the select logic selects ones of said plurality of instruction streams for which to fetch instructions for those instruction streams whose associated queues are least full.

15. (Previously Presented) The processor of claim 1 wherein by providing said plurality of instruction queues to decouple said fetch logic from said dispatch stage, and by providing said select logic to monitor said queues and select ones of said plurality of instruction streams for fetching, the processor allows a large number of streams in the processor to execute while minimizing the number of ports provided by said instruction cache.

16. (Previously Presented) The processor of claim 15 wherein said plurality of instruction queues is at least four, and said ports provided by said instruction cache is greater than one, but less than four.

17. (Previously Presented) The method of claim 7 further comprising:

providing dispatch logic, coupled to the plurality of instruction queues, the dispatch logic dispatching a variable number of instructions from each queue in each cycle.